42. (Amended) A computer system, comprising:

a bus;

a central processing unit coupled to said bus; and

a graphics accelerator coupled to said bus, said graphics accelerator including a texture cache system, said texture cache system including a texture cache memory that stores texels to be used by a texel value generating circuit, a cache controller that performs a replacement policy determination for texel data to be stored in said texture cache memory, and a direct memory access engine that retrieves texel data from memory.

E1

43. (Amended) The computer system of claim 42, wherein said texture cache memory is fully associative.

44. (Amended) The computer system of claim 42, wherein said replacement policy determination is performed in accordance with a least recently loaded policy.

45. (Amended) The computer system of claim 42, wherein said replacement policy determination operates such that cache lines containing texels that are being used to compute texture values to describe a polygon cannot be overwritten until said polygon is complete.

E1 Cont

46. (Amended) The computer system of claim 45, wherein said replacement policy determination is performed using at least one set of flags that are associated with cache lines of said texture cache.

52. (Amended) A texture mapping method using a graphics accelerator, said graphics accelerator including a texture cache memory, a cache controller, and a direct memory access engine, comprising:

(a) retrieving texels from memory via the direct memory access engine;

(b) storing said retrieved texels in the texture cache memory based on a replacement policy determination that is performed by the cache controller; and

E2

(c) rendering a polygon using texels that are stored in the texture cache memory.

53. (Amended) The texture mapping method of claim 52, wherein said storing comprises storing said retrieved texels in a fully associative texture cache memory.

54. (Amended) The texture mapping method of claim 52, wherein said replacement policy determination is performed in accordance with a least recently loaded policy.

55. (Amended) The texture mapping method of claim 52, wherein said replacement policy determination operates such that cache lines containing texels that are

E2 Cont

being used to compute texture values to describe a polygon cannot be overwritten until said polygon is complete.

70. (Amended) A graphics processing apparatus, comprising:

a graphics accelerator, said graphics accelerator, including

a texture cache memory;

a direct memory access engine coupled to a bus, wherein texel data is retrieved, by said direct memory access engine, from a memory over said bus and provided to said texture cache memory for storage; and

E3 Cont

a texture engine coupled to said texture cache memory; said texture engine receiving texels from said texture cache memory to produce texture values for pixels,

wherein storage in said texture cache memory is based on a replacement policy determination that is performed by a texture cache controller resident on said graphics accelerator.

71. (Amended) A graphics accelerator for use in a computer system having a central processing unit, comprising:

a texture cache memory;

a direct memory access circuit that retrieves texel data from memory;

a texture value generating circuit coupled to said texture cache memory; said texture value generating circuit producing texture values for pixels based on texel data stored in said texture cache memory; and

a cache controller that performs a replacement policy determination for said texture cache memory.

72. (Amended) The graphics accelerator of claim 71, wherein said texture cache memory is fully associative.

E3 Cont

73. (Amended) The graphics accelerator of claim 71, wherein said replacement policy determination is performed in accordance with a least recently loaded policy.

74. (Amended) The graphics accelerator of claim 71, wherein said replacement policy determination operates such that cache lines containing texels that are being used to compute texture values to describe a polygon cannot be overwritten until said polygon is complete.

75. (Amended) The graphics accelerator of claim 74, wherein said replacement policy determination is performed using at least one set of flags that are associated with cache lines of said texture cache.

Cont E3

76. (Amended) The graphics accelerator of claim 71, wherein said direct memory access circuit implements a virtual-physical address translation.

Please add new claims 90-99 as follows:

--90. The graphics accelerator of claim 70, wherein said texture cache memory is fully associative.

91. The graphics accelerator of claim 70, wherein said replacement policy determination is performed in accordance with a least recently loaded policy.

92. The graphics accelerator of claim 70, wherein said replacement policy determination operates such that cache lines containing texels that are being used to compute texture values to describe a polygon cannot be overwritten until said polygon is complete.

93. The graphics accelerator of claim 92, wherein said replacement policy determination is performed using at least one set of flags that are associated with cache lines of said texture cache.

94. The graphics accelerator of claim 70, wherein said direct memory access engine implements a virtual-physical address translation.

95. The graphics accelerator of claim 70, wherein said texture cache system is capable of operating in a prefetch mode such that during the rendering of a first

polygon, a set of texels including at least those texels needed for completely rendering a second polygon are prefetched and stored in said texture cache memory.

96. The graphics accelerator of claim 95, wherein said set of texels are prefetched if it is determined that said set of texels can fit into space available in said texture cache memory.

97. The graphics accelerator of claim 96, wherein said set of texels are prefetched if it is determined that said set of texels can fit into one half of said texture cache memory.

E4
Cont

98. The graphics accelerator of claim 95, wherein said texture cache operates in an on demand mode for said second polygon if said set of texels cannot be prefetched.

99. The graphics accelerator of claim 70, wherein said graphics accelerator is a graphics accelerator board.--

REMARKS

By the foregoing Amendment, claims 62-69 and 82-89 have been canceled, claims 42-46, 52-55, and 70-76 have been amended and new claims 90-99 have been added. These changes are believed not to introduce new matter, and their entry is respectfully requested.

Based on the above Amendment and the following Remarks, Applicants respectfully request that the Examiner reconsider all outstanding rejections, and that they be withdrawn.

Rejections under 35 U.S.C. §112, first paragraph

At paragraph 3 of the Office Action, the Examiner rejected claims 42-61 and 70-81 as containing subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the art that the inventor(s), at the time the application was filed, had possession of the claimed invention. In particular, the Examiner appears to object to the term "cache controller."

Applicants note that a "cache controller" is described in Applicants' specification at pages 19-20. Further, Applicants note that the terms "controller" and "control circuit" were used, for example, in originally filed claims 4 and 16, respectively. Applicants' use of the term "cache controller," which references a component resident on the graphics accelerator, is believed to be consistent with the prior usage of the term. Accordingly, for at least these reasons, Applicants submit that Applicants had possession of the claimed "cache controller" at the time the present application was filed.

Rejections under 35 U.S.C. §103

Claims 42-46, 48-56, 58-61, 70-75, and 77-81

At paragraphs 6-15 of the Office Action, the Examiner rejected claims 42-46, 48-56, 58-61, 70-75, and 77-81 as being unpatentable over U.S. Patent No. 5,790,130 to Gannett ("Gannett") and U.S. Patent No. 6,097,402 to Case et al. ("Case").

Gannett teaches a graphics system comprising a host 15 including a main memory 17 and a processor 19, a front end board 10, a texture mapping board 12, and a frame buffer board 14. See FIG. 4 of Gannett. Texture mapping board 12 is asserted as being equivalent to Applicants' graphics accelerator. As the Examiner noted:[1]

> Gannett further teaches a texture mapping board 12 that includes texture mapping chip 46, texel cache access circuit 82 comprising four controllers, and cache memory 48 ("a cache controller" and "a texture cache memory") that stores texture MIP map data downloaded from the main memory 17 associated with the primitives being rendered at col. 13 lines 44-55. Gannett also teaches that a texture interrupt managing daemon 160 that replaces cache blocks based on the least recently used blocks and the low priority blocks ("a replacement policy") at col. 42 lines 6-36.

In this excerpt, the Examiner explicitly refers to "a cache controller," "a texture cache memory," and "a replacement policy." By way of example, these terms are set forth in Applicants' amended claim 42 as follows:

[1] See paragraph 7 of the Office Action (Paper No. 22).

42. A computer system, comprising:
a bus;
a central processing unit coupled to said bus; and
a graphics accelerator coupled to said bus, said graphics accelerator including a texture cache system, said texture cache system including a texture cache memory that stores texels to be used by a texel value generating circuit, a cache controller that performs a replacement policy determination for texel data to be stored in said texture cache memory, and a direct memory access engine that retrieves texel data from memory.

As set forth above, Applicants' claim 42 recites "a cache controller that performs a replacement policy determination for texel data to be stored in said texture cache memory." Applicants submit that Gannett's texture mapping board does not include a controller that operates in a manner as recited in claim 42.

Gannett's controller is embodied in texture mapping chip 46, which is resident on texture mapping board 12. FIG. 7 of Gannett provides a detailed description of the relationship between texture mapping chip 46 and memory 48. The components of FIG. 7 operate as follows:[2]

> FIG. 7 is a block diagram of a cache memory implementation according to one illustrative embodiment, coupled to portions of the texture mapping chip including the texel port 92, texture interpolator 76, cache directory 78 and the texel cache access circuit 82. In this illustrative embodiment, the cache memory 48 includes four interleaves 204A, 204B, 204C and 204D. Each interleave includes two SDRAM chips (not shown) that can be accessed simultaneously, with each providing eights bits of data during a read cycle. . . . The texel cache access circuit 82 includes four separate controllers labeled controller A (200A), controller B (200B), controller C (200C) and controller D (200D). The four controllers A, B, C and D can simultaneously access data from the four interleaves 204A, 204B, 204C and 204D through parallel buses 202A, 202B, 202C and 202D. The

[2] See col. 21, lines 32-63 of Gannett.

> controllers read texel data from the memory 48 in response to commands and at addresses respectively received over buses 84A, 84B, 84C and 84D.

As applied by the Examiner, the four controllers 200A-200D are considered to be equivalent to Applicants' claimed "cache controller that performs a replacement policy determination for texel data to be stored in said texture cache memory." This equivalence, however, is not supportable.

Gannett's four controllers 200A-200D do not perform a replacement policy determination. Rather, as noted by the Examiner, the replacement policy determination is performed by the texture interrupt manager (TIM) daemon 160.

TIM 160 is an independent, stand-alone software process that runs on the processor 19 of host computer 15.[3] Upon a cache miss (i.e., situation where the corresponding texels are not stored in cache memory 48 when accessed by texture mapping chip 46), texture-mapping chip 46 generates an interrupt control signal. This interrupt control signal is provided over signal line 94 to distributor chip 30, which in turn provides an interrupt signal over line 95 to host computer 15.

The interrupt signal that is received by host computer 15 is processed by TIM 160. If cache memory 48 is full when the cache miss occurred, then one of the cache blocks is replaced. As Gannett states:[4]

> [W]hen a miss occurs for a block of texel data that is not in the cache, the TIM downloads to the cache 48 (FIG. 4) the requested block of texture data. If the cache was full when the miss occurred, then one of the cache blocks is replaced by the newly downloaded block of texture data. In one

[3] See FIG. 3B and col. 8, lines 51-54 of Gannett.
[4] See col. 42, lines 6-15 of Gannett.

> embodiment of the invention, the **software daemon determines which block to replace** by considering which blocks were used least recently and which blocks have texture data of low priority. (Emphasis added.)

As this excerpt demonstrates, TIM 160 is responsible for determining which block in cache memory 48 to replace. Gannett's replacement policy determination is therefore performed by TIM 160, not by a cache controller included on a graphics accelerator.

As Gannett further describes, TIM 160 uses a set of registers on texel port 92, which is included within texture mapping chip 46 of texture mapping board 12. This set of registers is read by TIM 160 through 3-D bypass path 24 and is used in determining which cache block to replace.[5] After the cache replacement policy determination is performed by TIM 160, the texel data is transferred to texture mapping chip 46 via 3-D bypass path 24. The received texel data is then stored in cache memory 48 using controllers 200A-200D.

As thus described, Gannett's controllers 200A-200D do not perform a replacement policy determination for texel data to be stored in cache memory. Rather, the replacement policy determination is performed by TIM 160. TIM 160 is not resident on texture mapping board 12. Instead, TIM 160 is resident on host computer 15.

For at least these reasons, Applicants submit that Gannett does not disclose a graphics accelerator that includes "a cache controller that performs a replacement policy determination for texel data to be stored in said texture cache memory." As the Examiner asserts that Gannett teaches all elements of claim 42 except for a DMA engine, the rejection of claim 42 is therefore traversed.

---

[5] See FIGS. 20-21 and col. 42, line 58 to col. 46, line 50 of Gannett.

Case is relied upon by the Examiner for the alleged teaching of an equivalent DMA engine. Even if Gannett teaches all that the Examiner alleges, Applicants submit that the combination of Gannett and Case is improper.

Case appears to describe a processing system that includes a graphics subsystem. The graphics subsystem further includes a graphics controller. The processing system of Case is designed to operate in various modes including a Direct Memory Execute (DIME) mode and a Direct Memory Execute and Local (DIMEL) mode.

In the DIME mode, graphics controller 141 of graphics subsystem 140 is capable of directly operating on texture maps from memory subsystem 130. Without DIME, processor subsystem 120 would fetch graphics information from memory subsystem 130 and write these texture maps to a dedicated, local memory 142 of graphics subsystem 140. This would produce unnecessary, redundant copies of the texture maps. Redundant copies of the texture maps are therefore avoided in DIME mode.[6]

In the DIMEL mode, graphics controller 141 is capable of retrieving frequently-used graphics information from dedicated local memory 142 as well as retrieving less-frequently used graphics information from memory subsystem 130.[7]

In the rejection, the Examiner states that it would have been obvious to combine the teachings of Gannett and Case "such that textures are downloaded using DMA for the purpose of directly transferring the texture data to the cache at a faster rate and without burdening the host CPU. Applicants submit that the Examiner has not set forth a proper motivation to combine the teachings of Gannett and Case.

---

[6] See col. 2, lines 55-67 of Case.
[7] See col. 3, lines 1-8 of Case.

First, Applicants note that Case does not disclose a DMA engine being used in a system that supports a texture cache. In Case, the DIME and DIMEL modes of operation enable the graphics controller to access either local memory or system memory. No interaction with a texture cache is described.

Second, Applicants further note that Gannett and Case use fundamentally different approaches to data transfer. In Gannett, data transfer is interrupt-driven and is controlled by the TIM software daemon that is running on the CPU of the host system. In Case, on the other hand, data transfer is controlled by a hardware graphics controller on a graphics subsystem.

Applicants submit that the Examiner has not set forth any basis for combining the teachings of Gannett and Case. Indeed, inclusion of the teachings of Case into the system of Gannett would require a fundamental change in the approach of Gannett in the control and use of a texture cache. Applicants' submit that these fundamental differences have not been addressed by the Examiner in setting forth the obviousness rejection. The Examiner has merely stated that it would have been obvious to combine the teachings of Gannett and Case "such that textures are downloaded using DMA for the purpose of directly transferring the texture data to the cache at a faster rate and without burdening the host CPU. Applicants request that the Examiner set forth in detail how the teachings of Case would be incorporated into the teachings of Gannett.

For at least those reasons state above, Applicants submit that the rejection of independent claim 42, as well as independent claims 52, 70, and 71, is traversed. As claims 43-46, 48-51, 53-56, 58-61, 72-75, and 77-81 are dependent either directly or

indirectly from one of independent claims 42, 52, and 71, the rejection of those claims is similarly traversed.

Claims 47, 57, and 76

At paragraphs 16-17 of the Office Action, the Examiner rejected claims 47, 57, and 76 as being unpatentable over Gannett, Case, and U.S. Patent No. 5,926,187 to Kim. As claims 47, 57, and 76 are dependent either directly or indirectly from one of independent claims 42, 52, and 71, the rejection of those claims is traversed for at least the reasons stated above.

Claims 62, 63, 65-68, 82-84, 86, 87, and 89

At paragraphs 18-21 of the Office Action, the Examiner rejected claims 62, 63, 65-68, 82-84, 86, 87, and 89 as being unpatentable over Gannett in view of International Publication WO 90/09634 to Olday et al. ("Olday"). By the amendment above, Applicants have canceled claims 62, 63, 65-68, 82-84, 86, 87, and 89. The rejection of claims 62, 63, 65-68, 82-84, 86, 87, and 89 is therefore rendered moot. Applicants reserve the right to argue the merits of claims 62, 63, 65-68, 82-84, 86, 87, and 89 in future continuation applications.

Claims 64, 69, 85, and 88

At paragraphs 22-23 of the Office Action, the Examiner rejected claims 64, 69, 85, and 88 as being unpatentable over Gannett, Olday, and U.S. Patent No. 5,945,997 to Zhao et al. By the amendment above, Applicants have canceled claims 64, 69, 85, and